

IN THE SPECIFICATION:

Please amend the paragraphs of the specification as set forth below. The first paragraph is a new paragraph, other paragraphs are replacement paragraphs for paragraphs in the specification as identified below.

New paragraph to be inserted on page 1, line 1:

This application is a continuation of U.S. Patent Application Serial No. 09/418,097, filed October 14, 1999.

Replacement Paragraph for the paragraph beginning on page 7, line 29:

Turning now to Fig. 1, a block diagram of one embodiment of a processor 10 is shown. Other embodiments are possible and contemplated. In the embodiment of Fig. 1, processor 10 includes a line predictor 12, an instruction cache (I-cache) 14, an alignment unit 16, a branch prediction/fetch PC generation unit 18, a plurality of decode units 24A-24D, a predictor miss decode unit 26, a microcode unit 28, a map unit 30, a retire queue 32, an architectural renames file 34, a future file 20, a scheduler 36, an integer register file 38A, a floating point register file 38B, an integer execution core 40A, a floating point execution core 40B, a load/store unit 42, a data cache (D-cache) 44, an external interface unit 46, and a PC silo 48. Line predictor 12 is coupled to predictor miss decode unit 26, branch prediction/fetch PC generation unit 18, PC silo 48, and alignment unit 16. Line predictor 12 may also be coupled to I-cache 14. I-cache 14 is coupled to alignment unit 16 and branch prediction/fetch PC generation unit 18, which is further coupled to PC silo 48. Alignment unit 16 is further coupled to predictor miss decode unit 26 and decode units 24A-24D. Decode units 24A-24D are further coupled to map unit 30, and decode unit 24D is coupled to microcode unit 28. Map unit 30 is coupled to retire queue 32 (which is coupled to architectural renames file 34), future file 20, scheduler 36, and PC silo 48. Architectural renames file 34 is coupled to future file 20. Scheduler 36 is coupled to register files 38A-38B, which are further coupled to ~~each other and~~ respective execution cores 40A-40B. Execution cores 40A-40B are further coupled to load/store

unit 42 and scheduler 36. Execution core 40A is further coupled to D-cache 44. Load/store unit 42 is coupled to scheduler 36, D-cache 44, and external interface unit 46. D-cache 44 is coupled to register files 38. External interface unit 46 is coupled to an external interface 52 and to I-cache 14. Elements referred to herein by a reference numeral followed by a letter will be collectively referred to by the reference numeral alone. For example, decode units 24A-24D will be collectively referred to as decode units 24.

Replacement Paragraph for the paragraph beginning on page 10, line 12:

Decode units 24A-24D decode the instructions provided thereto, and each decode unit 24A-24D generates information identifying one or more instruction operations (or ROPs) corresponding to the instructions. In one embodiment, each decode unit 24A-24B24D may generate up to two instruction operations per instruction. As used herein, an instruction operation (or ROP) is an operation which an execution unit within execution cores 40A-40B is configured to execute as a single entity. Simple instructions may correspond to a single instruction operation, while more complex instructions may correspond to multiple instruction operations. Certain of the more complex instructions may be implemented within microcode unit 28 as microcode routines (fetched from a read-only memory therein via decode unit 24D in the present embodiment). Furthermore, embodiments employing non-CISC instruction sets may employ a single instruction operation for each instruction (i.e. instruction and instruction operation may be synonymous in such embodiments).

Replacement Paragraph for the paragraph beginning on page 16, line 26:

The generated ROPs are written into scheduler 36 during the write scheduler stage. Up until this stage, the ROPs located by a particular line of information flow through the pipeline as a unit. However, subsequent to be written into scheduler 36, the

ROPs may flow independently through the remaining stages, at different times. Generally, a particular ROP remains at this stage until selected for execution by scheduler 36 (e.g. after the ROPs upon which the particular ROP is dependent have been selected for execution, as described above). Accordingly, a particular ROP may experience one or more clock cycles of delay between the write scheduler write stage and the read scheduler stage. During the read scheduler stage, the particular ROP participates in the selection logic within scheduler 36, is selected for execution, and is read from scheduler 36. The particular ROP then proceeds to read register file operations from one of register files 38A-38B (depending upon the type of ROP) in the register file read stage.

Replacement Paragraph for the paragraph beginning on page 17, line 29:

Turning now to Fig. 3, a block diagram illustrating one embodiment of branch prediction/fetch PC generation unit 18, line predictor 12, I-cache 14, predictor miss decode unit 26, an instruction TLB (ITLB) 60, an adder 62, and a fetch address mux 64 is shown. Other embodiments are possible and contemplated. In the embodiment of Fig. 3, branch prediction/fetch PC generation unit 18 includes a branch predictor 18A, an indirect branch target cache 18B, a return stack 18C, and fetch PC generation unit 18D. Branch predictor 18A and indirect branch target cache 18B are coupled to receive the output of adder 62, and are coupled to fetch PC generation unit 18D, ~~line predictor 12,~~ and predictor miss decode unit 26. Fetch PC generation unit 18D is coupled to receive a trap PC from PC silo 48, and is further coupled to ITLB 60, line predictor 12, adder 62, and fetch address mux 64. ITLB 60 is further coupled to fetch address mux 64, which is coupled to I-cache 14. Line predictor 12 is coupled to I-cache 14, predictor miss decode unit 26, adder 62, and fetch address mux 64.

Replacement Paragraph for the paragraph beginning on page 40, line 10:

Viewed in another way, the termination conditions for predictor miss decode unit

26 in creating line predictor entries are flow control conditions for line predictor 12. In other words, line predictor 12 identifies a line of instructions in response to each fetch address. The line of instructions does not violate the conditions of table 134, and thus is a line of ~~instruction~~ instructions that the hardware within the pipeline stages of processor 10 may be designed to handle. Difficult-to-handle combinations, which might otherwise add significant hardware (to provide concurrent handling or to provide stalling and separation of the instructions flowing through the pipeline) may be separated to different lines in line predictor 12 and thus, the hardware for controlling the pipeline in these circumstances may be eliminated. A line of instructions may flow through the pipeline as a unit. Although pipeline stalls may still occur (e.g. if the scheduler is full, or if a microcode routine is being dispatched, or if map unit 30 does not have rename registers available), the stalls hold the progress of the instructions as a unit. Furthermore, stalls are not the result of the combination of instructions within any particular line. Pipeline control may be simplified. In the present embodiment, line predictor 12 is a flow control mechanism for the pipeline stages up to scheduler 36. Accordingly, one microcode unit is provided (decode unit 24D and MROM unit 28), branch prediction/fetch PC generation unit 18 is configured to perform one branch prediction per clock cycle, a number of decode units 24A-24D is provided to handle the maximum number of instructions, I-cache 14 delivers the maximum number of instruction bytes per fetch, scheduler 36 receives up to the maximum number of instruction operations per clock cycle, and map unit 30 provides up to the maximum number of rename registers per clock cycle.

Replacement Paragraph for the paragraph beginning on page 53, line 4:

Turning now to Fig. 22, a block diagram of one embodiment of predictor miss decode unit 26 is shown. Other embodiments are possible and contemplated. In the embodiment of Fig. 22, predictor miss decode unit 26 includes a register 190, a decoder 192, a line predictor entry register 194, and a termination control circuit 196. Register 190 is coupled to receive instruction bytes and a corresponding fetch address from alignment unit 16, and is coupled to decoder 192 and termination control circuit 196.

Decoder 192 is coupled to line predictor entry register 194, to termination control circuit ~~192~~ 196, and to dispatch instructions to map unit 30. Line predictor entry register 194 is coupled to line predictor 12. Termination control circuit 196 is coupled to receive branch prediction information from branch predictors 18A-18C and is coupled to provide a branch address to fetch PC generation unit 18D and a CAM address to line predictor 12. Together, the branch prediction address, the CAM address, and the line entry (as well as control signals for each, not shown) may comprise the line predictor update bus shown in Fig. 3.

Replacement Paragraph for the paragraph beginning on page 58, line 27:

Processing nodes 312A-312D, in addition to a memory controller and interface logic, may include one or more processors. Broadly speaking, a processing node comprises at least one processor and may optionally include a memory controller for communicating with a memory and other logic as desired. More particularly, a processing node 312A-312D may comprise processor 10. External interface unit 46 may ~~includes~~ include the interface logic 318 within the node, as well as the memory controller 316.